**Composite Design Pattern**

The Composite design pattern is a structural pattern that allows you to compose objects into tree structures to represent part-whole hierarchies. This pattern enables clients to treat individual objects and compositions of objects uniformly. It is particularly useful when you need to work with tree-like structures.

**Definition**

The Composite pattern lets you treat individual objects and compositions of objects uniformly by allowing you to compose objects into tree structures. This pattern provides a way to work with individual objects and collections of objects (composite objects) in the same manner.

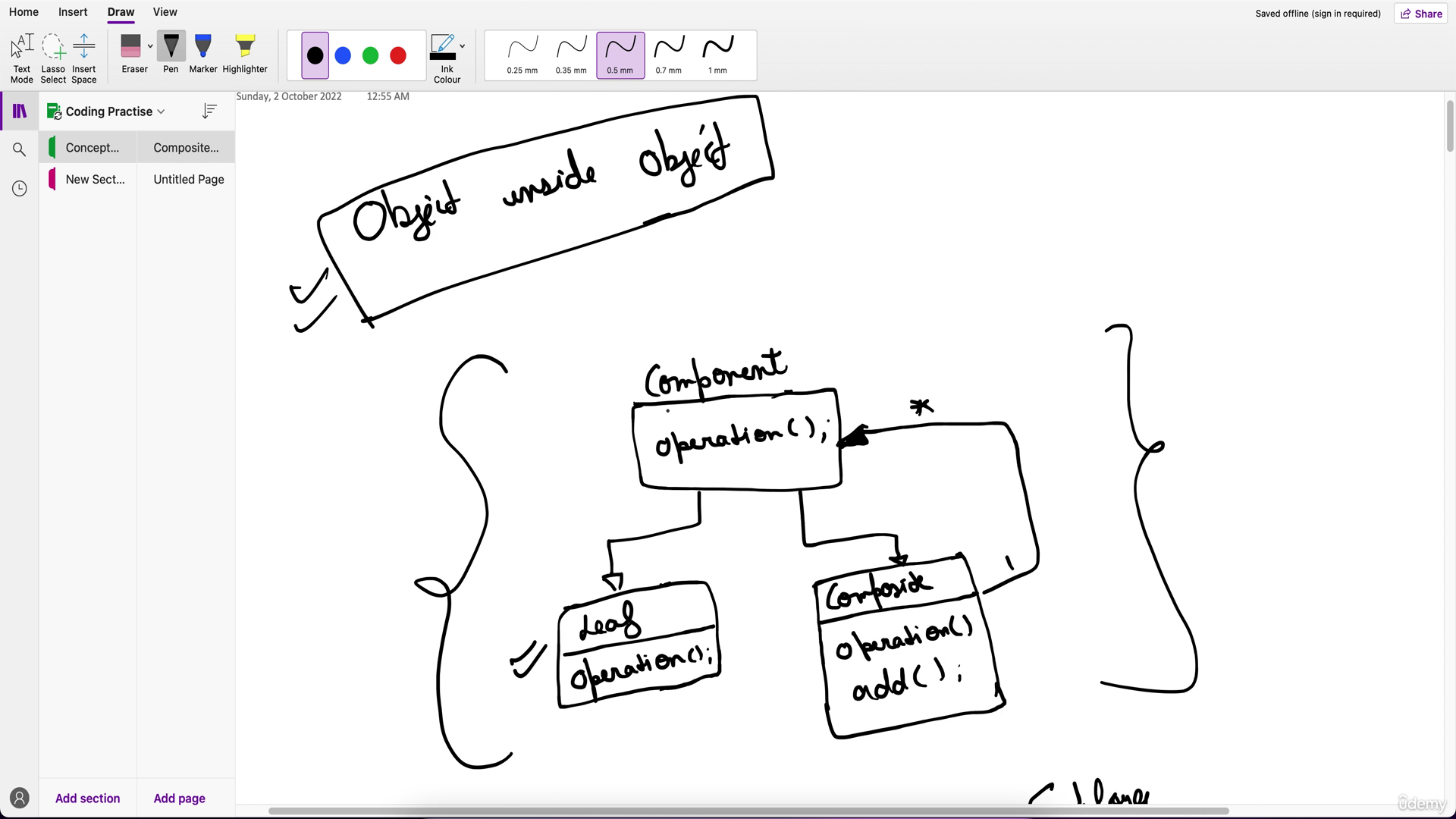
**Example**

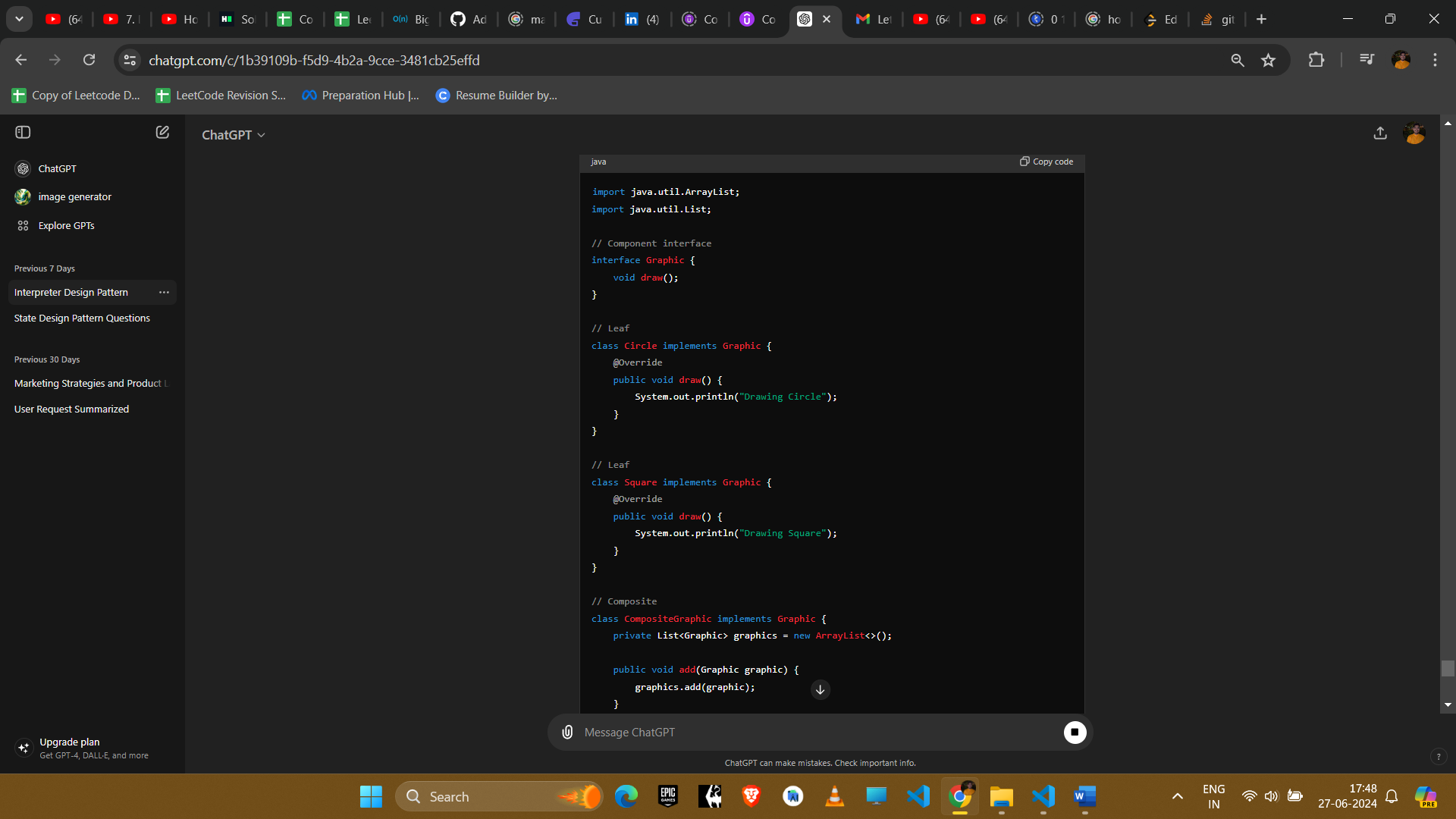
Consider a scenario where we have graphic objects, such as shapes, that can be grouped together to form complex structures.

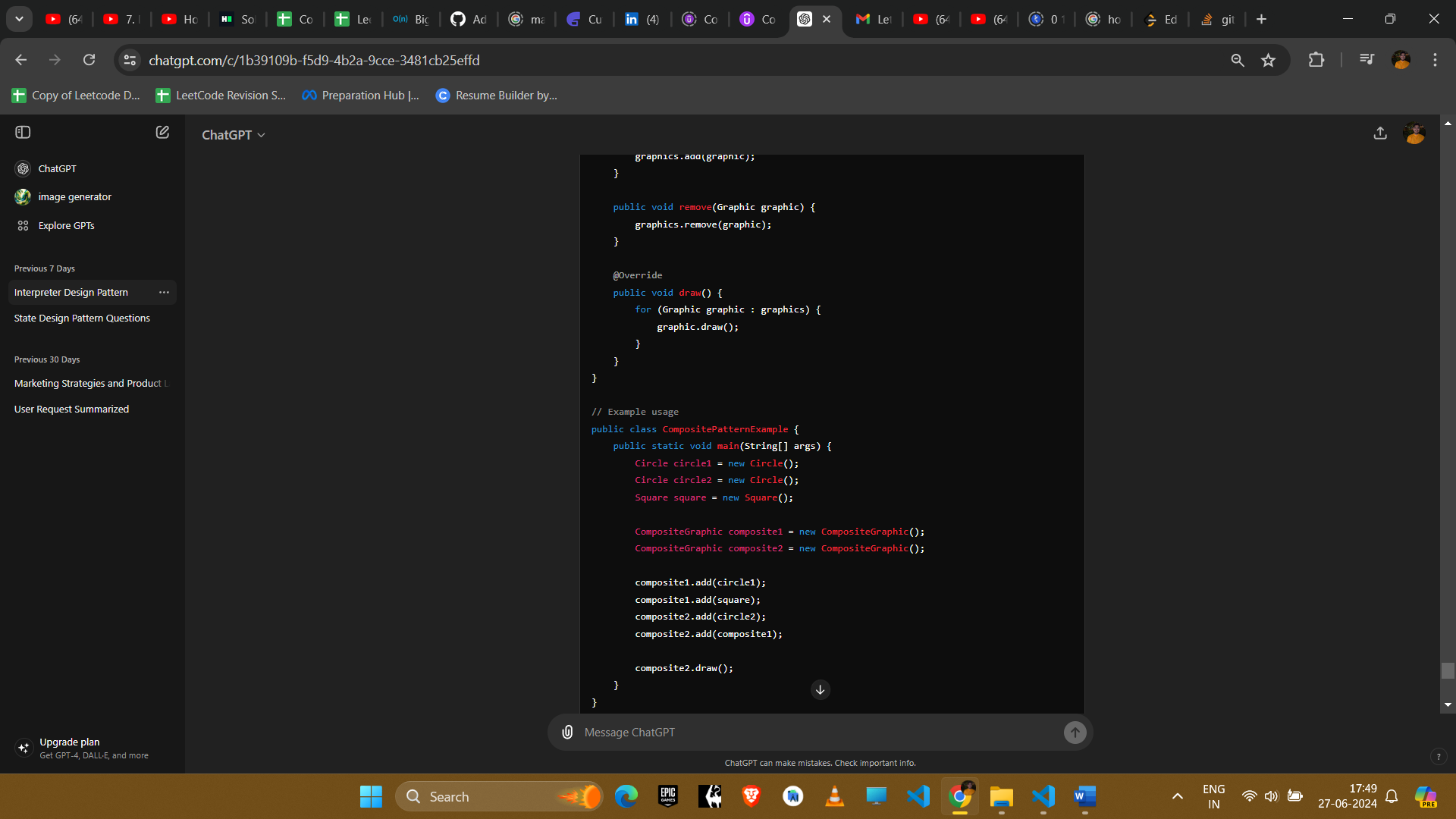
File system structure, hierarchy management, arithmetic expressions

**When to use:** when the problem appears like a tree structure

The Composite design pattern solves the problem of treating individual objects and compositions of objects uniformly. It allows you to build a tree structure of objects where individual objects and groups of objects are treated the same way, making it easier to work with complex hierarchical structures.







**Explanation**

1. **Component Interface**: Graphic defines the interface for all objects in the composition, both individual objects (leaves) and composite objects.
2. **Leaf**: Circle and Square are leaf components that implement the Graphic interface and represent individual objects.
3. **Composite**: CompositeGraphic is a composite component that implements the Graphic interface and contains a list of Graphic objects (both leaves and other composites).

**Example Uses in Amazon Interviews**

1. **File System Structure**
   * **Scenario**: Representing files and directories where directories can contain files and other directories.
   * **Implementation**: Use composite pattern to treat files and directories uniformly, allowing easy traversal and manipulation.
2. **Graphics Drawing Applications**
   * **Scenario**: Representing complex graphical objects composed of simpler shapes.
   * **Implementation**: Use composite pattern to compose shapes into complex graphical objects and render them uniformly.
3. **UI Component Hierarchies**
   * **Scenario**: Representing UI components such as panels, buttons, and text fields in a hierarchical structure.
   * **Implementation**: Use composite pattern to manage and render nested UI components in a consistent manner.
4. **Organization Structures**
   * **Scenario**: Representing organizational hierarchies where managers can have subordinates who are also managers or employees.
   * **Implementation**: Use composite pattern to manage and traverse the organizational hierarchy uniformly.

**Conclusion**

The Composite pattern is useful for representing part-whole hierarchies and allows clients to treat individual objects and compositions of objects uniformly. It is particularly beneficial in scenarios involving tree-like structures, providing a flexible and scalable solution for complex compositions.